29

Popular Computing

AUGUST 1975

VOLUME 3

NUMBER 8

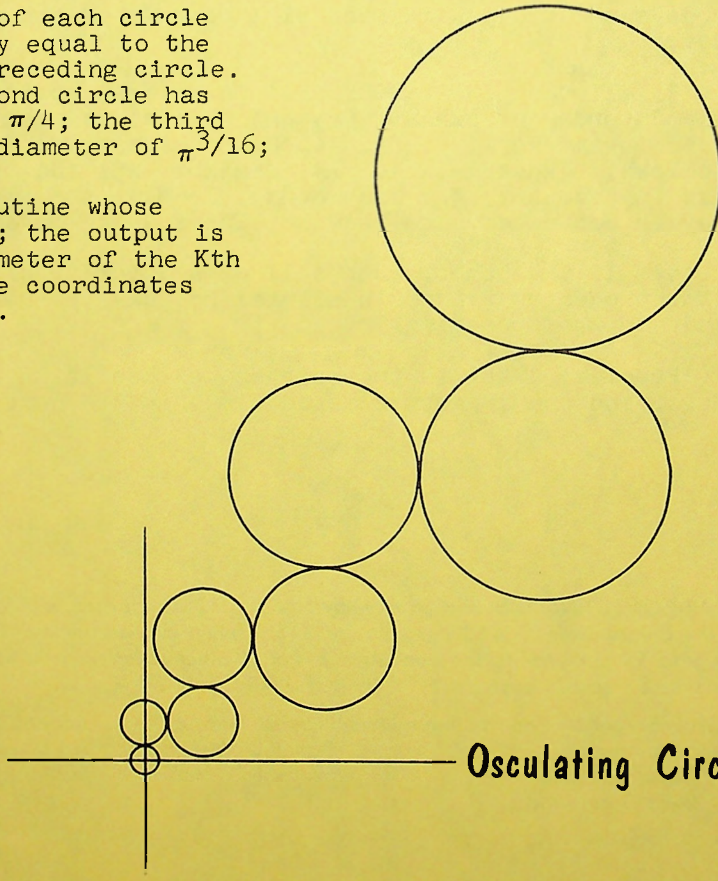
PROBLEM 96

A circle of unit diameter is centered at the origin.

Successive circles are drawn, each tangent to the preceding circle as shown. The points of tangency are alternately on the right and on top.

The diameter of each circle is numerically equal to the area of the preceding circle. Thus, the second circle has a diameter of $\pi/4$; the third circle has a diameter of $\pi^3/16$; and so on.

Write a subroutine whose argument is K; the output is to be the diameter of the Kth circle and the coordinates of its center.



Osculating Circles

The Payday Problem

A firm pays its employees twice a month, according to the following schedule:

1. Pay checks will normally be issued on the 5th and 20th of each month, if those days are working days.

2. If the 5th or 20th fall on weekends (that is, on Saturdays or Sundays) payment will be made on the preceding Friday.

3. If the 5th or 20th fall on a legal holiday, other than a Saturday or Sunday, payment will be made on the next following working day. Legal holidays are defined as follows: January 1, July 4, December 25, the third Monday in February, the last Monday in May, the first Monday in September, and the last Thursday in November.

4. If the 5th or 20th fall on a Saturday or Sunday, and the preceding Friday is a legal holiday, payment will be made on the preceding Thursday.

Problem: What is the shortest elapsed time between paydays, and the longest elapsed time, in the next 50 years? ☐

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $18 per year, or $15 if remittance accompanies the order. For Canada and Mexico, add $4 per year to the above rates. For all other countries, add $6 per year to the above rates. Back issues $2 each. Copyright 1975 by POPULAR COMPUTING.

The Altair 8800

The Altair 8800 computer, made by MITS, Inc., P.O. box 8636, 6328 Linn, N.E., Albuquerque, New Mexico 87108, was announced around the first of this year. The basic machine, with 256 8-bit words of storage, sells for $542 in kit form, or $755 assembled. The basic machine has volatile storage. Input is by means of toggle switches; output is through binary lights.

The machine we worked with had 1024 words of storage; this size is $615 in kit form or $830 assembled. The promotional literature says that 256 words of storage is "enough memory to do many machine language procedures and some control applications." This should be taken rather cautiously; there are few real problems whose solution can be programmed in that size machine.

The operation code length of the Altair is a full word of 8 bits. An instruction, therefore, consists of one or more words. For example, the operation ADD ONE TO ACCUMULATOR is a one-word instruction (octal 074). ADD IMMEDIATE takes two words:

```
306
037
```

(add decimal 31 to the contents of the accumulator). Most of the normal instructions that one expects on a computer take three words, such as:

```
072
126
002
```

for which the 072 is LOAD ACCUMULATOR, and the other two words indicate address 1126 octal, as follows:

```
00 000 010 | 01 010 110
0   0   2  | 1   2   6
```

(In the basic machine with 256 words of storage, the third of those three instruction words would always be zero.)

The 8800 uses the Intel 8080 chip as its central processor, and hence the Altair uses the op-codes of the Intel chip. Many of these are of marginal use in a general-purpose machine, such as RETURN (from subroutine) ON ZERO, RETURN ON MINUS, RETURN ON PLUS, and the like. Simple instructions like ADD take four words:

```
041
123
001
206
```

(add contents of word 523 to the accumulator).

The machine has a HALT instruction (166), but on execution it tends to lock the entire machine with all console lights on and all switches inoperative. This condition can be relieved by hitting STOP and RESET simultaneously. For purposes of halting execution, a better combination is:

```
A      303
A+1      A
A+2    ---
```

(unconditional jump to current location) which ties the machine into a short loop, during which the manual STOP switch can be used.

The machine is quite reliable, but its circulating storage is not protected by parity checks, and the loss or gain of a single bit can be disastrous.

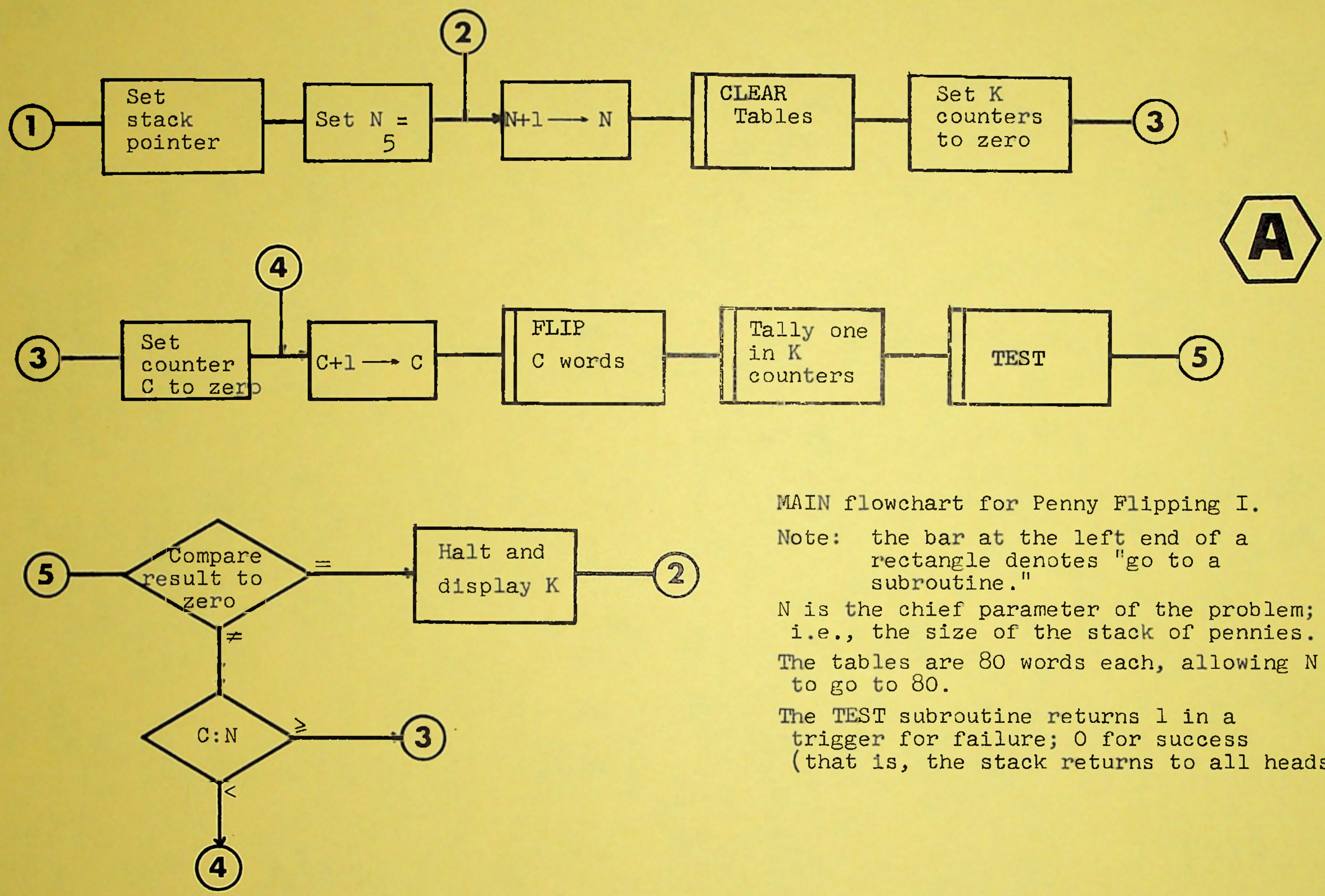
Promotion for the machine has leaned toward over-enthusiasm and hyperbole. For example, "It has 78 basic machine instructions with variances up to 200 instructions. That's enough to program all the traffic lights in a major city." The relation between the number of op-codes and any specific application is obscure.

Or, "Number of subroutine levels: 65,000." To call a subroutine takes 3 words; to return from a subroutine takes one word, and the return address (two words) must be stored somewhere. Thus, even with a full size machine with 65,000 words, it is difficult to determine just what the advertised statement could mean.

Additionally, the reference manual is in need of extensive re-writing; the explanations of the actions for the various op-codes are poorly written and hard to decipher even by experienced programmers.

(from the manual) "...the Altair 8800 can execute a six instruction addition program approximately 30,000 times in one second." This is true, but implies that the operating speed is 180,000 instructions per second; for a typical program, it will be more like 116,000 instructions per second.

Again, "Zero bit--this bit is set to 1 if the result of <u>certain instructions</u> is zero..." (underscores mine). The user of a computer wants to know the precise action of every op-code, and those things that can affect any bit.



MAIN flowchart for Penny Flipping I.

Note: the bar at the left end of a rectangle denotes "go to a subroutine."

N is the chief parameter of the problem; i.e., the size of the stack of pennies.

The tables are 80 words each, allowing N to go to 80.

The TEST subroutine returns 1 in a trigger for failure; 0 for success (that is, the stack returns to all heads).

The original Penny Flipping problem seems ideally suited to the machine. Given a stack of N pennies, all heads up. The top penny is turned over, then the top two pennies, the top three, and so on until the entire stack is turned over, after which the top one is turned over, the top two, and so on. The number of flips is counted, and the result is the functional value for that N. The results for various values of N are as follows:

N	f	N	f
1	2	11	121
2	3	12	119
3	9	13	116
4	11	14	195
5	24	15	75
6	35	16	79
7	28	17	204
8	31	18	323
9	80	19	228
10	60	20	199

Flowcharts are given for this problem, to extend the function from 21 to 80. Since the counts of the number of flips can go over 127 (the limit of the size of a positive number in one word), a two-word counter is set up, and subroutine C shows the logic of incrementing those counters (the scheme readily extends to larger counters).

We begin by allocating storage as follows:

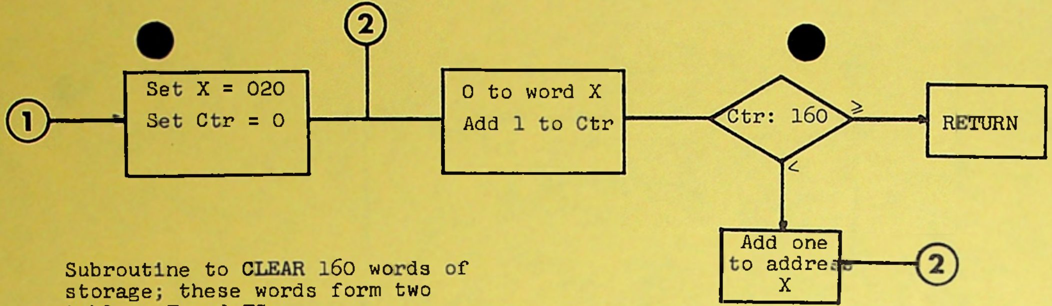
Table T, words 020 through 137 octal (this table will contain the stack of pennies, one penny per word).
Table TC (T copy), words 140 through 257 octal.
K counters, words 002 and 003.
N, word 004.
C, word 005.
Temporary storage, word 006.
Trigger (for TEST), word 010.
Counter, word 011.

The MAIN flowchart (A) shows the overall logic of the problem. The program begins (as must all programs on the Altair that use subroutines) by setting the stack pointer:

061
377
003

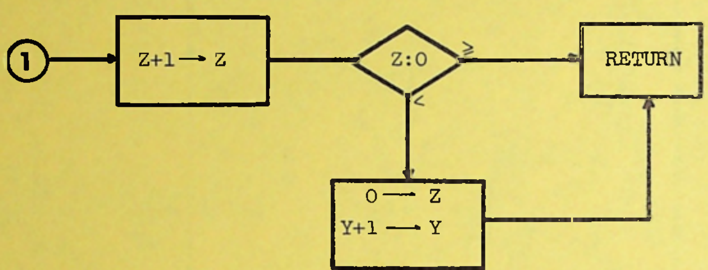
to provide a place in storage (in this case at octal 1777) for stacking return addresses for all subroutines.

N will be initialized to 6, at Reference 2. Tables T and TC are cleared to zero. The flowchart for this subroutine (B) shows a loop that counts to 160. This is possible, but it might be easier (if this is your first approach to the machine) to form two loops, each counting to $80)_{10}$.



B

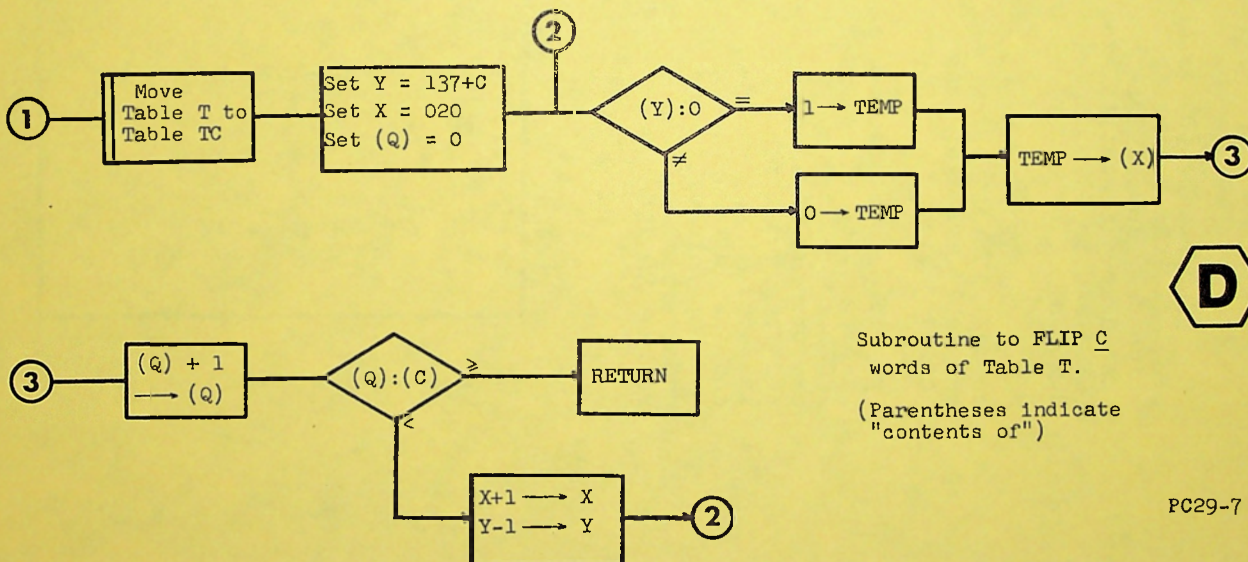
Subroutine to CLEAR 160 words of storage; these words form two tables, T and TC.



word 2	word 3
Y	Z

C

Subroutine for K COUNTERS. Two words of storage are used to form a counter with capacity up to 128^2.



D

Subroutine to FLIP C words of Table T.

(Parentheses indicate "contents of")

Subroutine (D) (FLIP) performs the heart of the problem. It begins by calling on another subroutine (for which the flowchart is not given) to move table T to TC. Then a loop is initialized to the first address of Table T (020) and the last address of Table TC (137 + C), to flip C words. The counter Q is word 011.

The TEST subroutine (E) examines all of Table T for zeros. The flipping task is completed when N words of the table are again zero, but it is simpler to test the entire table.

The complete program, written in straightforward fashion (that is, not using tricky coding) occupies about 400 words of storage, plus some 170 words of data. It executes at an average rate of 80 flips per second, so that the total run for N = 74 is 70 seconds (the result is 5475). For each value of N, the machine is halted, and the contents of the counters is displayed. For N = 74, this will read as 052 and 143; these results in octal are changed to 42 and 99 (decimal), and the result is then 42·128 + 99 = 5475.

Further results have also been obtained for the Penny Flipping IV problem (see PC25-12) as follows:

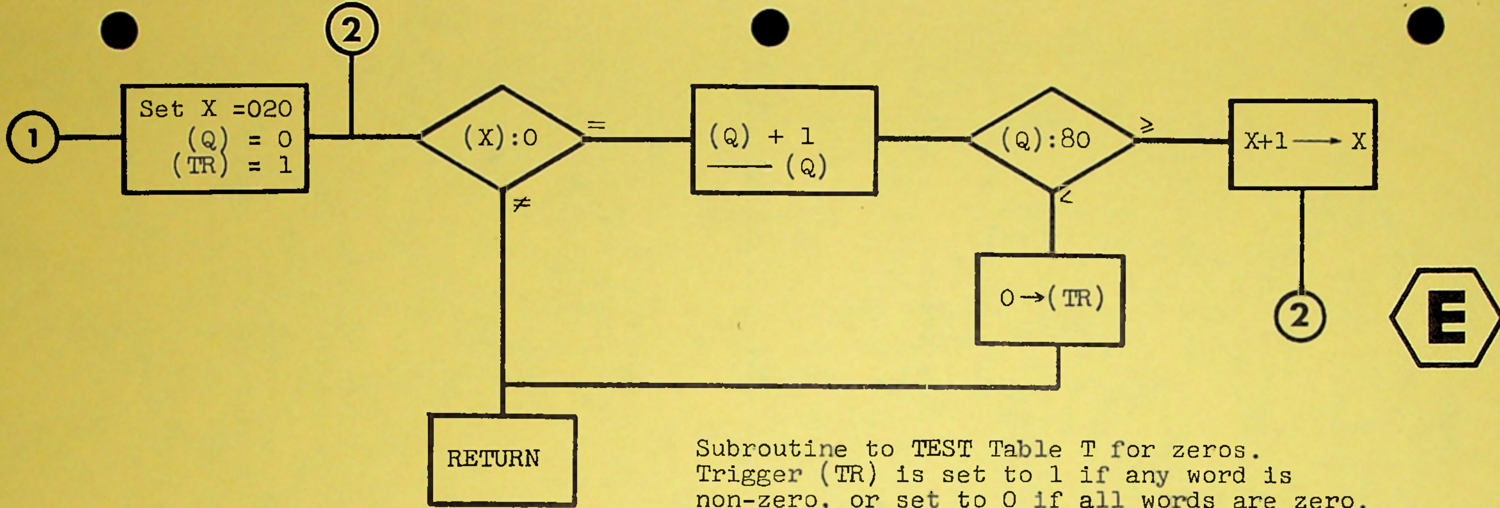
Allan Herschderfer and Charles McCord, April 1975, on a CDC 3170

N	f
38	163540
39	20064
40	31122
41	51040
42	11480
43	16128
44	450296

FJG, May 1975, on the Altair 8800

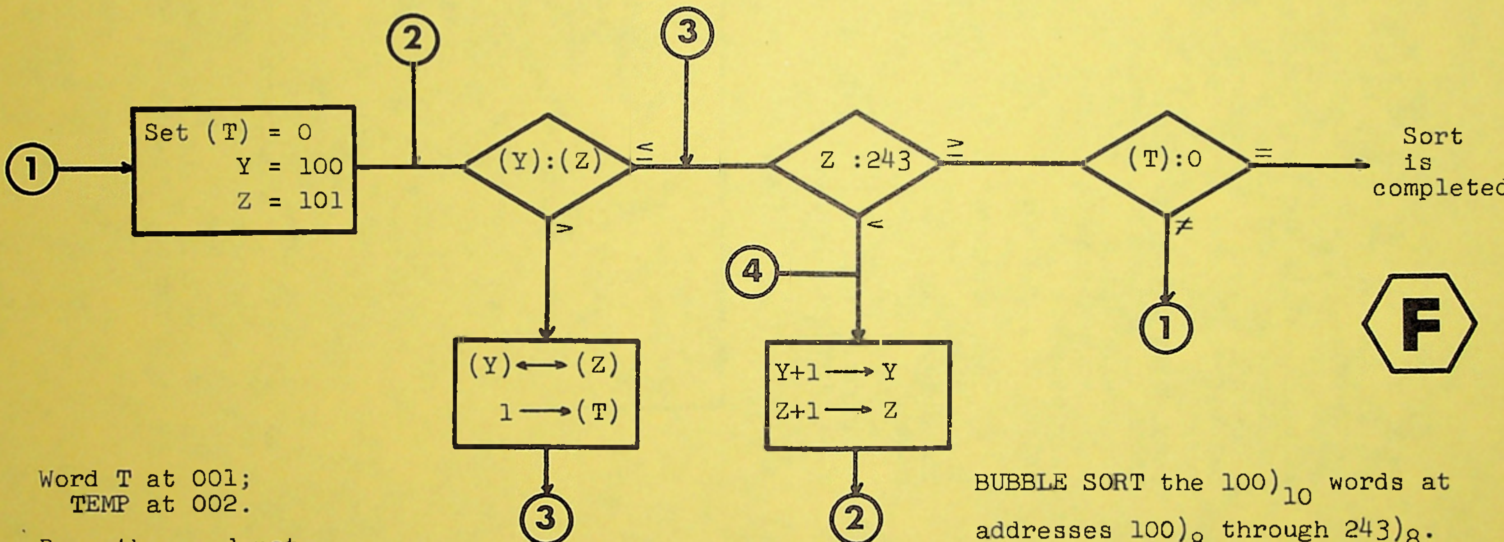
N	f
45	88685
46	4860
47	15088
48	518880
49	43776
50	25284
51	82000
52	5100
53	73008
54	2270520
55	517083

For another example of the level that most users of the Altair 8800 will work at, consider the task of sorting one hundred numbers. The given data is stored in words $100)_8$ through $243)_8$, and we will produce a machine language code to bubble sort it.



Subroutine to TEST Table T for zeros. Trigger (TR) is set to 1 if any word is non-zero, or set to 0 if <u>all</u> words are zero.

Note: The loop examines all of Table T.



Word T at 001;
TEMP at 002.

Parentheses denote "contents at address."

BUBBLE SORT the $100)_{10}$ words at addresses $100)_8$ through $243)_8$.

Bubble (or interchange) sorting is considered by many (including Knuth) to be a dirty technique that should not be encouraged in any way. Granted that bubble sorting is crude, inelegant, unsophisticated, and (for more than a few items to be sorted) highly wasteful of machine time, it still exists and has definite virtues. For one, it is simple and easy to code in any language, and hence easy to debug and test. Its inherent inefficiency hurts you only if the number of things to be sorted is large (say, over 30) or the number of sets of those things is large (say, over 1000), or both. Otherwise, the programmer is choosing to swap machine time (and possibly storage space) for brain strain, or for a reduction in elapsed time--and that swap can be profitable. Finally, the technique is widely used, and there should be no hidden or suppressed tricks in our trade.

The accompanying flowchart (F) shows the logic of the bubble sort, together with a complete code for it (G). The code, like the technique itself, is simple, using no esoteric coding tricks, nor any of the hidden registers of the Altair. If one chooses to use an inefficient technique, then it is inconsistent to try to use it in a sophisticated way.

PROBLEM 98

Finally, a problem is offered that is particularly suited to the Altair 8800, but is solvable on any machine. The accompanying diagram (H) shows 32 8-bit words. The words are indexed at the left, from 00 through 37 octal. The individual bits in the words are also indexed at the top, from 000 through 111 binary. The heavy line divides each word into two parts. The 5-bit part is the address of one of the 32 words; the 3-bit part addresses one of the bits of that word. For example, the word at 00000 addresses the 5th bit from the left in word 10011 (that bit is circled in the diagram).

The 32 words are to be scanned in turn, and each indicated bit is to be changed, from 1 to 0, or from 0 to 1. The implementation of this simple situation is a nice exercise in bit manipulation. The Problem is: Will the process converge for any starting combination of bits (other than the obvious case in which all 256 bits are initially zero)? □

Fred Gruenberger

```
500   076   R-1
501   000
502   062   Set (T) = 0
503   001
504   000
505   076
506   100
507   062
510   134
511   001   Set Y=100
512   062
513   151
514   001
515   062
516   162
517   001
520   076
521   101
522   062
523   137
524   001   Set Z=101
525   062
526   157
527   001
530   062
531   170
532   001
533   072   R-2
534   000 Y
535   000   (Y) - (Z)
536   041
537   000 Z
540   000
541   226
542   312   To R-3
543   177
544   001   on zero
545   372
546   177   or minus
547   001
550   072
551   000
552   000   (Y) → (TEMP)
553   062
554   002
555   000
556   072
557   000
560   000   (Z) → (Y)
561   062
562   000
563   000
```

```
564   072
565   002
566   000   (TEMP) → (Z)
567   062
570   000
571   000
572   076
573   001
574   062   1 → (T)
575   001
576   000

577   072   R-3
600   137
601   001
602   326   Z: 243
603   243
604   372   Go R-4
605   222
606   001
607   072
610   001
611   000
612   306   (T): 0
613   000
614   312
615   257
616   001
617   303   Go R-1
620   100
621   001
622   072   R-4
623   134
624   001
625   074
626   062   Y+1 → Y
627   134
630   001
631   062
632   151
633   001
634   062
635   162
636   001
```

```
637   072
640   137
641   001
642   074
643   062
644   137   Z+1 → Z
645   001
646   062
647   157
650   001
651   062
652   170
653   001
654   303   Go R-2
655   133
656   001

657   Continue
```

Code for the Altair 8800 to bubble sort the words from $100\text{-}243)_8$ in ascending order.

	000	001	010	011	100	101	110	111
00000	1	0	0	1	0	0	1	1
00001	1	0	1	1	1	1	1	0
00010	1	1	0	0	0	0	0	0
00011	1	0	0	1	1	1	1	1
00100	0	0	1	1	1	0	0	0
00101	1	1	0	0	1	1	1	1
00110	1	1	0	0	1	1	0	0
00111	1	0	0	0	0	0	1	1
01000	1	1	1	0	0	0	0	1
01001	1	0	0	0	1	1	1	0
01010	1	1	0	0	1	0	0	1
01011	1	1	0	1	0	1	1	0
01100	1	0	1	0	1	0	0	0
01101	1	1	1	0	0	1	0	1
01110	0	1	1	0	1	1	0	1
01111	0	1	1	0	0	0	0	1
10000	0	1	0	1	0	1	1	0
10001	1	1	1	1	1	1	0	1
10010	0	1	0	0	0	0	1	1
10011	1	0	1	0	1	0	1	0
10100	0	0	1	0	1	0	0	0
10101	0	1	0	1	1	0	0	1
10110	0	0	0	1	1	0	0	1
10111	1	1	0	1	1	0	1	1
11000	1	0	0	1	0	1	1	0
11001	1	1	0	1	0	1	0	0
11010	0	1	1	0	1	1	0	0
11011	0	0	1	1	1	1	1	1
11100	0	0	0	1	0	0	0	1
11101	0	1	0	0	0	1	0	0
11110	0	1	1	0	0	0	1	0
11111	1	1	1	1	0	1	0	1

NOTONE Research

The game of NOTONE was proposed by Walter Koetke in the first issue (Nov/Dec 1974) of Creative Computing. Two players take turns tossing two dice. When it is a player's turn, he tosses the dice and establishes his point. He may then toss the dice as many times as he wishes, and his score for the turn is the sum of the tosses. If his point reappears, however, his turn ends then with a score of zero. A game consists of ten rounds for each player.

A program (in BASIC) for the game was presented in the Mar/Apr 1975 issue, written by Robert Puopolo. The printout showed a score of 345 for the human player, indicating an average score of 34.5 per turn.

What is the proper strategy of play for this game?

If the point is 7, there is a .50 probability of making a run of 3 tosses before a 7 reappears. If the point is 2 (or 12) however, one can expect a run of 24 tosses before the reappearance of the 2 (or 12), again with a .50 probability. These probabilities are summarized in the accompanying table (A). For example, with a point of 5 (or 9), and a .45 probability, a run of 6 tosses can be expected. Thus, the attempted run at each turn is controlled by the point that is first made, as well as the probability level that will maximize the sum. Notice that although the probability of a run of a given length is the same for points 3 and 11, the expected sum is not. For example, if a .60 probability is used, the expected run for 3 or 11 is 9 tosses, but the 9 tosses starting with 11 will total higher than the 9 tosses starting with 3.

The point about dice is that they turn two flat distributions (that is, two distributions uniformly distributed in the range from 1 to 6) into a peaked distribution from 2 to 12 with these theoretical frequencies:

2	3	4	5	6	7	8	9	10	11	12
1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

PC29-14

B

8	8	10	6	2	8	2	6	7	8	12	3	11	7	9	7	8	7	3	11	8	11	7	6	4	11	8	11	8	4
6	9	7	9	9	11	7	5	5	7	6	7	6	7	7	6	6	7	4	12	10	5	6	8	5	4	6	9	8	4
10	9	5	8	7	5	11	9	7	9	4	6	7	8	7	10	5	10	7	5	5	6	10	8	11	6	11	6	5	7
6	10	7	3	6	5	4	6	2	8	4	8	3	12	8	8	5	8	7	5	10	4	4	5	10	7	9	5	6	8
5	8	11	6	5	3	5	8	7	7	8	5	11	10	7	4	4	10	7	3	8	4	7	6	6	8	4	5	11	7
6	8	5	2	8	9	3	8	3	6	9	10	3	8	7	7	7	5	8	12	7	12	4	7	9	10	8	8	9	9
5	9	12	5	8	9	6	4	3	6	7	6	7	6	8	8	7	11	8	9	9	11	10	8	8	8	8	5	3	11
4	8	3	11	7	4	8	7	10	6	3	2	3	8	7	4	7	6	4	6	10	8	6	10	5	5	6	6	9	8
12	7	3	8	8	12	8	7	8	8	7	6	5	6	12	9	4	6	5	7	6	7	11	8	8	7	8	5	4	10
9	3	9	8	9	4	6	3	5	9	7	6	5	7	7	7	7	7	9	3	8	9	11	11	11	7	8	3	7	7
12	2	4	10	5	6	8	7	8	7	3	9	6	7	8	7	7	9	6	10	10	9	7	9	7	8	11	5	9	11
10	6	10	9	6	9	6	7	8	9	7	8	5	7	9	8	3	10	6	5	7	10	5	3	6	8	6	8	8	8
9	6	8	8	9	9	7	5	9	6	11	12	7	3	5	8	8	6	3	10	6	7	3	8	7	3	5	7	5	7
10	7	8	7	9	6	10	4	7	5	8	4	12	2	8	9	4	8	7	7	9	8	7	7	8	8	7	6	7	4
9	12	5	6	8	8	3	4	4	8	9	9	4	7	4	5	7	3	2	4	10	7	3	6	6	9	2	10	12	9
4	5	6	9	10	11	8	10	9	6	7	5	5	5	6	7	9	9	11	5	10	11	10	8	7	6	5	5	8	8
7	6	10	9	8	7	4	10	8	5	7	4	10	4	7	7	9	9	7	7	4	3	6	8	11	3	2	5	5	7
8	8	7	10	7	7	7	7	9	6	9	7	10	2	8	3	8	7	10	6	11	5	9	12	10	10	7	5	8	11
7	7	5	7	8	11	8	12	5	7	6	3	8	6	5	8	8	7	8	12	6	6	3	6	9	5	2	6	3	7
5	5	3	5	10	6	6	11	9	9	8	9	8	10	5	10	10	8	7	3	6	10	9	6	8	6	5	8	9	6
5	7	11	8	3	8	6	8	5	2	7	10	2	5	5	10	4	6	8	3	2	6	6	5	6	7	10	9	7	7
9	8	7	11	7	7	11	3	5	9	12	9	4	9	10	2	5	4	6	3	7	7	5	12	7	10	4	8	6	10
7	7	10	5	7	8	11	5	4	9	3	8	9	7	10	7	8	8	6	5	8	4	10	6	8	4	9	8	6	9
8	7	5	5	6	9	8	7	7	4	7	5	7	2	7	5	4	6	7	9	6	9	7	8	11	7	6	10	8	8
7	9	9	7	6	4	8	8	8	10	3	6	7	3	11	8	11	6	4	11	11	5	8	6	9	4	8	7	8	6
7	8	7	6	10	10	6	11	9	3	3	3	8	3	10	7	5	6	8	7	7	5	9	6	3	10	10	3	4	5
8	10	11	7	5	7	10	9	5	6	6	5	7	6	2	8	6	7	6	5	8	10	6	10	12	8	10	8	7	8
5	4	7	10	8	5	4	8	8	9	2	7	3	4	5	7	4	2	6	11	3	7	8	7	12	6	12	11	9	10
7	6	4	7	7	5	7	7	12	7	5	8	8	5	9	6	12	7	4	8	5	6	11	6	8	6	11	5	6	9
4	7	10	10	3	10	10	8	4	9	2	6	6	10	3	5	8	4	7	7	7	5	6	6	6	8	7	10	8	4

6	7	6	8	10	11	7	10	8	9	9	3	5	9	12	9	8	10	4	5	12	7	9	4	6	6	5	10	4	9
9	9	8	11	8	9	3	10	10	9	6	9	9	8	8	6	5	6	5	8	9	11	11	6	8	11	3	6	7	2
5	7	2	11	8	9	9	4	3	8	8	8	4	8	4	6	10	7	8	5	8	6	2	11	11	6	5	4	10	7
4	3	5	6	6	6	12	4	6	8	7	5	9	4	5	5	9	9	8	6	3	2	7	12	6	6	6	4	7	6
8	8	7	4	12	4	6	9	8	6	7	9	7	4	6	8	8	10	7	7	9	6	5	6	10	7	9	7	4	9
9	3	10	9	5	10	12	4	11	7	4	6	6	3	8	10	9	10	4	9	4	4	9	3	11	11	7	9	6	6
9	5	9	5	4	9	11	5	5	9	2	5	7	9	9	3	10	7	7	7	7	2	3	5	7	4	5	10	12	7
5	7	8	6	8	9	8	5	12	8	6	9	11	7	9	9	8	10	5	5	5	5	8	10	5	6	9	3	4	5
4	11	7	4	8	12	4	6	5	9	8	7	4	3	4	5	9	9	4	2	6	9	7	8	11	8	9	7	6	3
6	6	4	8	5	10	8	12	7	9	9	9	12	3	7	7	8	5	9	6	11	10	4	5	6	6	6	7	8	11
6	6	10	9	5	4	8	7	9	9	6	7	7	9	10	10	5	7	7	10	9	4	3	3	7	6	4	7	5	7
5	9	8	10	11	9	6	4	4	5	5	8	7	7	5	4	9	6	10	3	4	8	7	10	5	8	6	8	4	12
7	8	8	6	7	9	7	11	8	10	8	4	8	8	12	3	12	11	5	7	9	9	11	3	12	5	6	7	9	8
6	5	7	7	4	9	10	10	8	7	10	9	9	5	8	3	9	6	7	7	10	8	7	6	10	9	10	5	11	8
10	8	3	10	8	5	11	3	3	6	5	12	6	5	8	7	6	5	6	9	7	7	3	7	5	5	3	9	7	7
8	8	8	7	10	7	9	9	9	4	5	11	9	3	9	7	9	2	6	7	5	4	8	10	10	10	6	8	11	7
6	9	5	5	6	10	7	8	8	7	7	6	6	5	9	10	10	4	2	7	7	8	7	4	4	9	9	8	6	5
5	10	4	2	7	8	12	5	5	11	7	7	6	6	2	4	8	3	10	5	7	4	4	6	11	8	5	11	8	6
3	8	3	7	8	7	5	7	2	6	10	7	5	3	3	7	5	9	7	7	11	10	9	6	10	7	7	5	4	9
6	2	5	7	10	10	6	4	8	12	6	10	8	10	7	6	11	9	6	6	2	7	7	10	8	10	9	7	7	6
6	10	9	7	11	6	4	6	5	7	8	6	5	6	7	9	8	6	3	9	6	6	9	11	6	5	6	3	9	9
12	4	6	4	6	9	10	9	9	10	5	7	7	5	6	6	11	6	8	7	6	4	7	11	11	6	6	11	10	8
8	11	5	6	8	6	11	10	8	7	11	7	7	10	6	12	5	7	5	7	10	11	10	11	9	6	7	5	3	6
6	11	7	9	11	6	9	10	8	9	8	10	6	9	10	9	8	8	7	9	8	8	12	4	4	7	10	10	6	5
11	6	9	3	8	6	8	8	8	2	4	8	9	7	9	9	6	5	9	4	7	9	10	7	4	6	10	9	4	7
9	6	7	9	4	9	10	9	4	11	8	6	5	8	9	8	5	11	8	4	8	10	9	6	4	4	6	6	4	6
9	11	5	9	3	8	3	7	6	8	6	12	7	4	9	7	7	5	5	8	5	7	7	5	11	10	9	5	9	5
10	6	6	8	6	7	8	3	7	3	5	11	5	9	5	11	8	7	8	10	8	9	3	7	5	5	8	11	7	6
5	5	7	8	9	8	6	8	2	10	9	7	9	5	7	4	3	6	7	9	8	6	7	7	7	9	5	7	10	9
9	9	6	6	7	8	7	12	7	3	8	6	12	8	11	10	6	9	9	10	5	3	4	5	9	8	8	5	8	5

	.80	.75	.70	.60	.50	.45	.40	.30
2	7	10	13	18	24	28	32	40
3	4	5	6	9	12	14	16	20
4	2	3	4	6	7	9	10	14
5	1	2	3	5	5	6	7	10
6	1	1	2	4	4	5	6	8
7	1	1	1	3	3	4	5	7

A

PC29-15

Most problems involving dice tosses (those that do not yield to an analytic solution) are best explored by computer. At times, however, some research can be done by hand, given access to suitable dice tosses. The accompanying table (B) gives 1800 tosses of two dice, for which the observed frequencies are as follows:

2	3	4	5	6	7	8	9	10	11	12
37	98	132	200	247	302	268	216	154	98	48

(the chi-squared value for goodness-of-fit is 8.432.)

In the traditional game of craps, the spots appearing on two dice are added. If they were multiplied, the tosses would take this distribution:

1	1/36	8	2/36	18	2/36
2	2/36	9	1/36	20	2/36
3	2/36	10	2/36	24	2/36
4	3/36	12	4/36	25	1/36
5	2/36	15	2/36	30	2/36
6	4/36	16	1/36	36	1/36

Suppose, then, we define the game of CRAPS2 as follows:

The player with the dice makes a toss. If he gets a square (1, 4, 9, 16, 25, or 36) he loses but retains the dice. If he throws 6 or 12, he wins. In all other cases, he has established his point and continues to throw. If he makes his point before the appearance of either 6 or 12, he wins; otherwise he loses.

The usual dice questions are then:

1. What are the odds against winning?

2. What are the odds against making the points of 2, 3, 5, 8, 10, 15, 18, 20, 24, and 30?

3. What are the expected lengths of runs between successive appearances of each of the 18 possible numbers?

Traditionally, dice are used by twos. Suppose they were tossed three at a time, with the spots added? The possibilities then range from 3 to 18 with these frequencies:

3	1/216	8	21/216	13	21/216
4	3/216	9	25/216	14	15/216
5	6/216	10	27/216	15	10/216
6	10/216	11	27/216	16	6/216
7	15/216	12	25/216	17	3/216
		18	1/216		

PROBLEM 100

The same questions (above) could then be raised for CRAPS3, played like normal craps but with these variations: On the first toss, totals of 10 or 11 are an immediate win; totals of 3 or 18 are an immediate loss. Otherwise, a point has been established, and the player keeps tossing until he makes his point (a win) or until 10 or 11 appears (a loss).

Finally, the game of NOTONE can be played with two (multiplicative) dice, or with three (additive) dice, and the proper strategy for these two games is to be determined.

* * * * * *

A computer run was made on tosses of two dice, to determine empirically the length of strings between successive appearances of each possible number. The following table shows the statistics gathered on 36000 tosses:

Point	N	Sum	Sum of squares
2	975	27278	1059380
3	2006	33753	984789
4	3000	35543	766037
5	4057	35850	598062
6	4984	35985	474381
7	5968	35978	393352
8	5094	36025	469881
9	3969	35907	615531
10	2999	35583	777941
11	1956	33940	1003532
12	992	27396	1055050

For example, the point 2 appeared 975 times; the sum of the lengths of the strings of tosses between successive 2's was 27278 and the sum of the squares of those string lengths was 1059380. Thus, the average run length between successive 2's was 27.977 and the standard deviation was 26.93.

Printed by Argold Press Inc.
16921 Ventura Boulevard, Encino

Combinatorial

PROBLEM **103**

There are 63 numbers that are made up of the factors 2, 3, 5, 7, 11, and 13, each taken no more than once. For example, eight of the 63 numbers are:

$$
\begin{aligned}
7 &= 7\\
39 &= 3 \times 13\\
143 &= 11 \times 13\\
154 &= 2 \times 7 \times 11\\
1365 &= 3 \times 5 \times 7 \times 13\\
2002 &= 2 \times 7 \times 11 \times 13\\
15015 &= 3 \times 5 \times 7 \times 11 \times 13\\
30030 &= 2 \times 3 \times 5 \times 7 \times 11 \times 13
\end{aligned}
$$

These 63 numbers total 96767; their mean is 1535.9841 and their standard deviation is 4210.7.

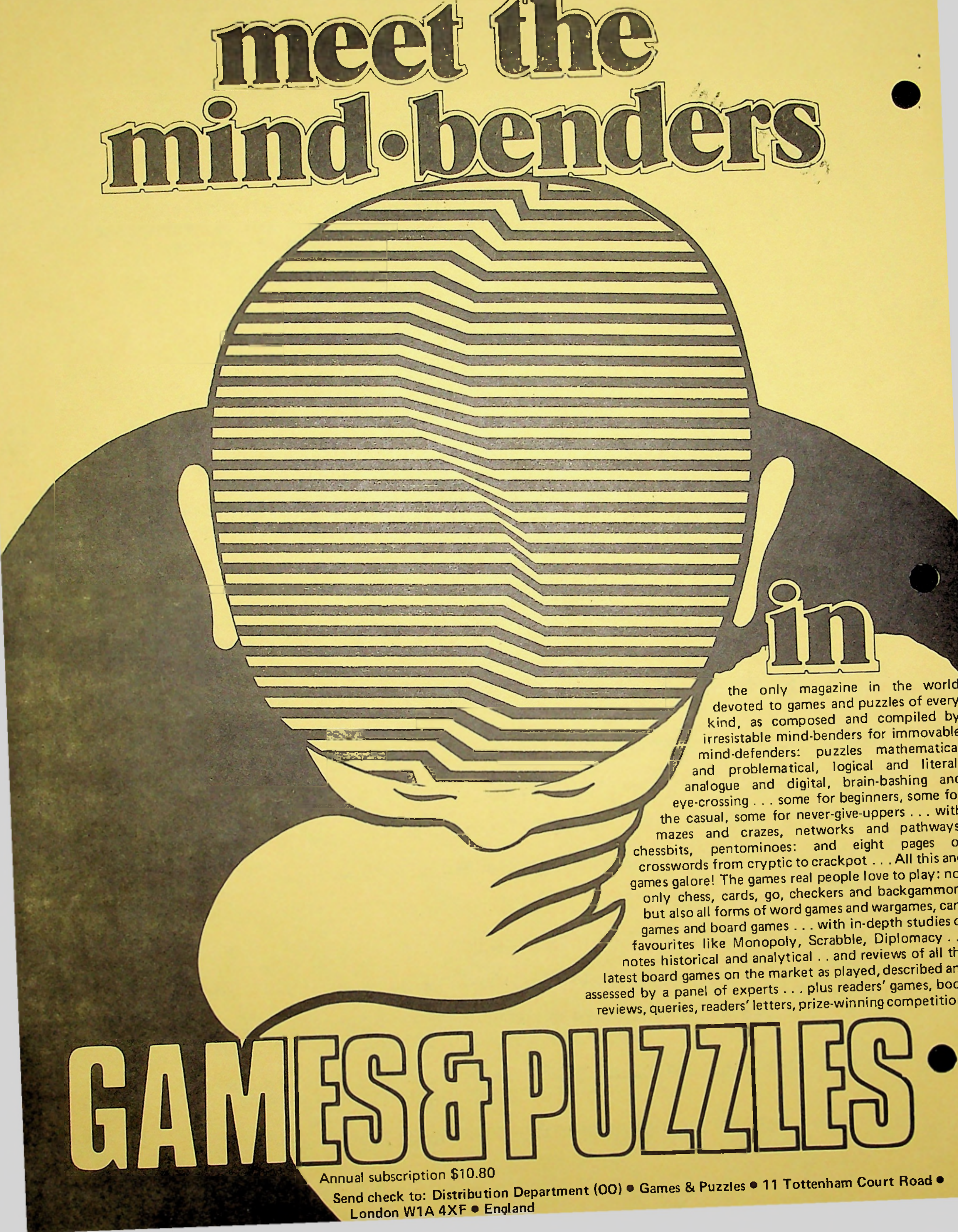
(1) What will be the sum, mean, and standard deviation for the 63 numbers similarly made up of the factors 3, 5, 7, 11, 13, and 17?

(2) Write a Fortran program that will utilize six factors (I, J, K, L, M, and N, all integers, all relatively prime to each other) and calculate the sum, mean, and standard deviation of the 63 numbers that can be formed.

* * * * * * * * * * * * * * * * * * *

Log 29	1.462397997898956087332846762969254991254294417887154
Ln 29	3.367295829986474027183272032361911605494512913922744
$\sqrt{29}$	5.385164807134504031250710491540329556295120161644789
$\sqrt[3]{29}$	3.072316825685847293312637982105597485502783238876096
$\sqrt[5]{29}$	1.961009057454548013206356850097824143767813931689090
$\sqrt[10]{29}$	1.400360331291395876495106634543574736747598468978608
$\sqrt[100]{29}$	1.034246309741025914720034059677913189875415336590493
e^{29}	3931334297144.042074388620580843527685796942333443902 185643090885541997881948228824965987972
π^{29}	261424513284460.8709628724332156234476313094693009254 5656390579769711436771704172403172432
$\tan^{-1} 29$	1.536327225795388605121153267285351612646716182990619

N-SERIES **29**



meet the mind·benders
in
the only magazine in the world devoted to games and puzzles of every kind, as composed and compiled by irresistable mind-benders for immovable mind-defenders: puzzles mathematical and problematical, logical and literal, analogue and digital, brain-bashing and eye-crossing . . . some for beginners, some for the casual, some for never-give-uppers . . . with mazes and crazes, networks and pathways, chessbits, pentominoes: and eight pages of crosswords from cryptic to crackpot . . . All this and games galore! The games real people love to play: not only chess, cards, go, checkers and backgammon, but also all forms of word games and wargames, card games and board games . . . with in-depth studies of favourites like Monopoly, Scrabble, Diplomacy . . . notes historical and analytical . . and reviews of all the latest board games on the market as played, described and assessed by a panel of experts . . . plus readers' games, book reviews, queries, readers' letters, prize-winning competitions
GAMES&PUZZLES
Annual subscription $10.80
Send check to: Distribution Department (OO) • Games & Puzzles • 11 Tottenham Court Road • London W1A 4XF • England